第 37 卷第 5 期
录用定稿
计算机应用研究
Application Research of Computers
ChinaXiv合作期刊
Vol. 37 No. 5
Accepted Paper

融合协同过滤的 XGBoost 推荐算法 *

齐德法 [1]，徐连诚 [1,†]，朱振方 [2]

(1. 山东师范大学 信息科学与工程学院，济南 250358; 2. 山东交通学院 信息科学与电气工程学院，济南 250357)

摘 要：在推荐系统中，针对用户的冷启动问题，提出一种融合协同过滤的 XGBoost 推荐算法。根据基于用户相似度的协同过滤推荐算法进行粗粒度召回，得到部分用户的召回集，使用 XGBoost 算法对召回集中的项目进行预测。对于存在冷启动问题的用户，直接使用 XGBoost 算法对候选集中的项目进行预测。该算法采用 CCIR2018 个性化推荐评测的在线评测数据集，并将推荐结果投放到知乎提供的线上平台进行评测。评测结果表明，该算法可以地解决用户的冷启动问题，具有很高的执行效率，准确度高，在线上评测中取得显著的推荐效果，并获得三等奖。

关键词：协同过滤；冷启动；XGBoost；推荐系统

中图分类号：TP301.6　　doi: 10.19734/j.issn.1001-3695.2018.10.0808

XGBoost recommendation algorithm with collaborative filtering

Qi Defa[1], Xu Liancheng[1,†], Zhu Zhenfang[2]

(1. *School of Information Science & Engineering*, *Shandong Normal University*, *Jinan* 250358, *China*; 2. *School of Information Science & Electrical Engineering*, *Shandong Jiaotong University*, *Jinan* 250357, *China*)

Abstract: In the recommendation system, this paper proposed an XGBoost recommendation algorithm to integrate collaborative filtering based on the cold-start problem of users. Firstly, it uses coarse grain to recall according to the collaborative filtering recommendation algorithm based on user similarity, and get a recall set of some users. Then using XGBoost algorithm to predict the items in the recall set. Secondly, for users with cold-start problems, it can directly use XGBoost algorithm to predict the items in the candidate set. Finally, the algorithm uses the online evaluation data set of CCIR2018 personalized recommendation evaluation, and puts the recommendation results on the online platform provided by Zhihu for evaluation. The evaluation results show that the algorithm in this paper can solve the cold-start problem of users with high efficiency and accuracy. It has achieved remarkable recommendation effect in the online evaluation platform and gets the third prize.

Key words: collaborative filtering; cold-start; XGBoost; recommender system

0　引言

随着互联网的飞速发展，信息资源呈现几何级数增长，丰富的网络信息给用户带来极大的便捷，使用户能够通过网络获取更多的知识信息。但数据规模的爆炸式增长[1]却产生了信息过载的问题，用户在大量的信息中很难获取到真正需要的数据信息。传统搜索引擎的出现解决了大数据时代初期信息过载的问题，但随着互联网的进一步发展，传统的搜索引擎已无法满足用户的特殊需求，因此个性化推荐系统应运而生，并成为了解决信息过载问题的有效方法[2]。个性化推荐系统能够根据用户个人喜好自动进行信息推荐，减少信息的冗余，简化用户的操作，从而带来更好的用户体验。

个性化推荐算法多种多样，传统的推荐算法主要有基于内容的推荐、协同过滤推荐[3]、混合推荐三种方式。基于内容的推荐是根据用户的历史交互记录，挖掘出与目标用户喜欢项目的相似项目。协同过滤是根据用户对项目的评分计算用户或项目之间的相似度，找出用户或项目的最近邻集合，最后产生推荐列表并进行推荐 [4]。混合推荐是将多种推荐算法按照一定的规则组合在一起，然后进行预测，最终将结果推荐给目标用户。

协同过滤是经典的推荐算法，其对推荐对象没有特殊要求，能够处理复杂的对象信息，可解释性强，简单易实现，因此协同过滤推荐算法被广泛应用。在实际应用场景中，协同过滤存在冷启动问题。冷启动问题是指在推荐系统中，缺少用户历史行为数据，系统无法得到用户的个人偏好，最终导致系统推荐效果较差。例如，当系统不存在或者存在少量的用户历史交互记录时，将无法得到目标用户的相似用户。同理，对于新项目也存在相同的冷启动问题。冷启动是协同过滤推荐算法中被广泛关注的问题，冷启动问题的存在严重影响了推荐系统的推荐质量[5]。

针对冷启动问题，陈克寒等人[6]提出一种基于两阶段聚类的推荐算法 GCCR，将图摘要方法和基于内容相似度的算法结合，实现基于用户兴趣的主题推荐；于洪等人[7]提出了用户时间权重信息概念，将时间权重信息应用到推荐系统中，可以判断该用户是积极用户还是消极用户，以及用户对新项目的偏爱程度，该算法在准确度和新颖度都有较好的效果；Pereira 等人[8]将用户人口统计信息引入到推荐算法，形成混合协同过滤推荐算法，可以很好地解决冷启动问题；

收稿日期：2018-10-28；**修回日期：**2018-12-13　**基金项目：**国家自然科学基金资助项目（61373148）；国家自然科学基金青年基金资助项目（61502151）；山东省社科规划项目（16CXWJ01，17CHLJ18，17CHLJ33，17CHLJ30）；山东省自然科学基金资助项目（ZR2014FL010）；山东省教育厅基金资助项目（J15LN34）

作者简介：齐德法（1993-），男，山东济宁人，硕士研究生，主要研究方向为个性化推荐；徐连诚（1973-），男（通信作者），副教授，硕导，主要研究方向为网络信息安全（lchxu@163.com）；朱振方（1980-），男，副教授，硕导，主要研究方向为自然语言处理.

Shambour 等人[9]在传统的基于用户的协同过滤推荐算法中，融入了项目评分信任度的思想，同时摈弃了传统的相似度计算方法，该算法既能缓解数据稀疏性问题，又能很好地解决冷启动问题。

本文算法旨在解决推荐系统中的冷启动问题，同时提高算法的执行效率，减少推荐过程所消耗的时间，提高模型的准确率，改善用户体验。为了减少模型的计算量，本算法首先使用基于用户的协同过滤推荐算法进行项目的召回，对于活跃用户，可以得到用户的召回集。召回集的项目数量远远少于候选集的项目数量，使用 XGBoost 算法在召回集上进行回归预测，降低了模型的计算量，提高了预测效率。对于历史交互记录稀疏或者缺失的用户，此类用户对应的召回项目较少，甚至没有召回项目。对于该类用户，使用 XGBoost 算法直接在全部候选集中进行预测，虽然原始候选集的数据量较多，但此类用户在推荐系统中所占的比例很低，因此消耗的时间较少。通过本文算法可以有效地缓解用户的冷启动问题，同时又可以减少模型计算所消耗的时间，提高算法的执行效率。通过 CCIR2018 清华大学与知乎联合举办的个性化推荐评测的在线评测，本文算法取得了显著的推荐效果。

1 相关工作

1.1 协同过滤

协同过滤[10]是经典的推荐算法，其在工业界得到了广泛应用。协同过滤主要分为基于用户的协同过滤、基于项目的协同过滤和基于模型的协同过滤。基于用户的协同过滤是根据用户的历史行为，挖掘出与目标用户相似的用户，然后向目标用户推荐其相似用户所喜好的项目；基于项目的协同过滤是根据项目的相似性，向用户推荐其喜欢项目的相似项目；基于模型的协同过滤首先根据训练集数据，采用概率统计模型或者机器学习方法建立模型（如潜在语义模型、贝叶斯模型、决策树模型、图模型等），进而通过模型预测目标用户的偏好[11]。

定义包含 m 个用户的集合 $U=\{u_1,u_2,\cdots,u_m\}$，包含 n 个项目的集合 $I=\{i_1,i_2,\cdots,i_n\}$，用户对项目的历史交互记录矩阵 S 可以表示为

$$S_{m\times n}=\{s_{11},s_{12},\cdots,s_{1n},\cdots s_{m1},s_{m2},\cdots,s_{mn}\}$$

基于用户的协同过滤和基于项目的协同过滤，都是基于相似度进行计算，根据用户对项目的历史交互记录矩阵 S，可以得到相似用户或者相似项目，最后得到推荐列表。本文相似度计算公式使用 Ochiai 系数[12]，该系数源于生物学中两个地区共同物种分布区域的相似度计算方法，是余弦相似度[13]的一种形式。

Ochiai 系数计算公式为

$$K=\frac{|A\cap B|}{\sqrt{|A|\times|B|}} \tag{1}$$

其中：A 和 B 表示集合；$|A|$ 和 $|B|$ 分别表示集合 A 和 B 中的元素个数。

1.2 XGBoost

XGBoost[14]是由陈天奇博士提出的 Gradient Boosting[15]算法，是 Gradient Boosting 的一种高效系统实现，其对 GBDT[16]进行了优化改进，能够并行计算、近似建树、对稀疏数据进行有效处理，对 CPU 和内存的使用进行了优化，使其在机器学习领域展现了很好的效果。

首先，XGBoost 在 GBDT 的基础上，通过在目标函数中加入正则化项，减小模型的复杂度，避免过拟合。其目标函数为

$$\mathcal{L}(\phi)=\sum_i l(\hat{y}_i,y_i)+\sum_k \Omega(f_k) \tag{2}$$

$$where\ \Omega(f)=\gamma T+\frac{1}{2}\lambda\omega^2$$

其中：$\hat{y}_i$ 为预测值；y_i 为真实值；l 为损失函数；$\Omega(f_k)$ 为正则项；f_k 为一棵决策树。XGBoost 算法还使用了二阶泰勒展开，假设第 t 次的损失函数为

$$\mathcal{L}^{(t)}=\sum_{i=1}^{n} l(y_i,\hat{y}_i^{(t-1)}+f_t(x_i))+\Omega(f_t) \tag{3}$$

对 $\mathcal{L}^{(t)}$ 做二阶泰勒展开

$$\mathcal{L}^{(t)}\simeq\sum_{i=1}^{n}\left[l(y_i,\hat{y}^{(t-1)})+g_i f_t(x_i)+\frac{1}{2}h_i f_t^2(x_i)\right]+\Omega(f_t) \tag{4}$$

$$where\quad g_i=\partial_{\hat{y}^{(t-1)}} l(y_i,\hat{y}^{(t-1)})$$

$$h_i=\partial^2_{\hat{y}^{(t-1)}} l(y_i,\hat{y}^{(t-1)})$$

其中：g_i 为一阶导数；h_i 为二阶导数。通过二阶泰勒展开，可以加快模型收敛速度，得到全局最优解。

XGBoost 有原始 XGBoost 库和 scikit-learn 库的两种实现方式，为用户提供了多种参数，通过调节不同的参数，得到最优模型。XGBoost 库中的基学习器不仅可以使用 CART，也可以使用线性分类器。其效率很高，在 Kaggle 等比赛中取得了很好的成绩，在工业界也得到了广泛的使用。

2 融合协同过滤的 XGBoost 推荐算法的模型构建

本模型使用协同过滤和 XGBoost 相融合的算法进行内容推荐。首先通过协同过滤得到用户的召回集；然后使用 XGBoost 模型对召回集中的项目进行预测，若不满足推荐数量，则继续使用 XGBoost 对候选集进行预测；最终产生推荐列表。模型构建主要分为三个阶段：首先是基于协同过滤的内容召回；其次是 XGBoost 模型训练；最后是产生推荐列表。算法流程如图 1 所示。

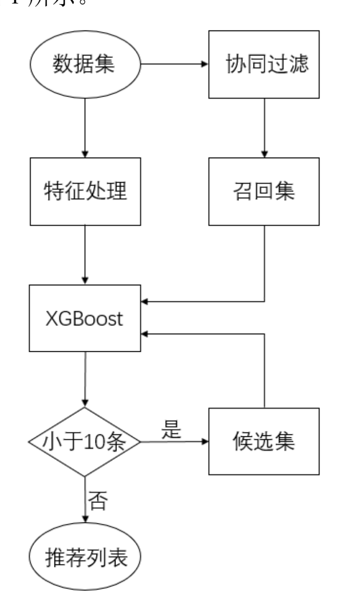


图 1 算法流程

Fig. 1 Flow of algorithm

阶段 1：基于协同过滤的内容召回。

输入：用户历史交互记录。

输出：部分用户的召回集。

算法步骤：

a) 在训练集中提取每个用户点击的项目。

b) 根据用户点击项目，计算用户之间的相似度。其相似度 S 计算公式为

$$S=\frac{n(I_{u_a}\cap I_{u_b})}{\sqrt{n(I_{u_a})\times n(I_{u_b})}} \tag{5}$$

其中：I_{u_a} 表示第 a 个用户的历史交互项目；I_{u_b} 表示第 b 个用户的历史交互项目；n 表示项目的数量。

c) 根据用户相似度，得到每个用户最相似的用户，然后将最相似用户已点击而该用户未点击的项目作为召回集，但因存在冷启动问题，该召回集仅为部分用户的召回集。

阶段 2：XGBoost 模型训练。

输入：用户的历史交互记录，用户的特征数据，项目的特征数据。

输出：XGBoost 模型。

算法步骤：

a) 提取用户历史交互数据，将存在点击记录的项目视为 1，不存在点击记录的项目视为 0。

b) 对数据进行均衡化，使已点击项目和未点击项目的比例为 1:1。

c) 提取用户的特征，并对特征进行预处理。对于文本型特征，将其转换为 one-hot 编码。对于时间戳型特征，截取前三位数字。对于数字型特征，需要进行离散化处理。

d) 使用 XGBoost 进行模型训练，并对参数进行调整，得到最终模型。

阶段 3：产生推荐列表。

输入：部分用户的召回集，原始候选集，XGBoost 模型。

输出：推荐列表。

算法步骤：

a) 使用 XGBoost 模型对部分用户的召回集进行预测，将预测值大于阈值的项目作为推荐结果。

b) 若推荐结果小于 10 条，则使用 XGBoost 在原始候选集中进行预测，并将预测值大于阈值的项目作为推荐结果。

c) 对于存在冷启动问题的用户，则将此类用户在原始候选集中进行预测，并得到推荐结果。

d) 将所有推荐结果进行去重处理，对于历史交互记录已存在项目，不再向用户进行推荐。

e) 为每个用户提取推荐结果中的前 10 条作为推荐列表，将结果投放到线上环境进行推荐。

3　实验结果与分析

3.1　实验数据与评测

本实验采用的数据集是 CCIR2018 个性化推荐评测的在线评测数据集，该数据集来自知乎移动端的信息流推荐数据。该数据集包括用户信息、内容信息、用户与内容的历史交互信息。其中用户信息包括注册时间、性别、访问频率、关注、提问次数、回答次数、终端设备、所在地等相关信息，内容信息包括内容对应的问题 id、是否匿名、是否被推荐、媒体信息、赞同次数、评论次数、收藏次数等相关信息，用户与内容的历史交互包括内容展示的时间、用户点击的时间、用户搜索的时间等信息。知乎每天给出 10 000 个用户的历史交互信息数据以及相关用户、内容的信息数据，同时给出候选集，在候选集中为每个用户分别选择 10 条内容作为推荐结果，从第二天凌晨开始将该结果投放到知乎线上环境进行评测。

3.2　数据预处理

在数据集中，不同维度的特征具有不同的量纲，在特征选择中，需要对数据进行无量纲化处理。通过将不同维度的特征全部转换为数字表示的特征，特征之间才能够进行计算，最终得到目标函数。

在评测数据集中，特征类型有多种形式，不同类型的特征，数据处理方式不同，主要分为时间戳类型特征、文本类型特征、数字类型特征。在 XGBoost 模型训练时，需要对特征数据进行预处理。其处理方式如表 1 和 2 所示。

表 1　用户数据特征处理

Table 1　Feature processing of user data

特征名称	特征样例	处理方式	结果样例
注册时间	1540475871	截取前 3 位	154
性别	男、女	one-hot	01、10
访问频率	daily、weekly	one-hot	001、010
关注用户数	238、25	等频划分	3、1
关注话题数	128、36	等频划分	4、1
被评论数	63、655	等频划分	1、10
被点赞数	48、190	等频划分	1、4
注册类型	other	one-hot	00010000
注册平台	android	one-hot	00000100
是否用 PC	0、1	不处理	0、1
设备 model	Mi8	one-hot	00000100
用户平台	Xiaomi	one-hot	00100000
用户所在省	北京	one-hot	00001000
用户所在市	北京	one-hot	00100000

表 2　内容数据特征处理

Table 2　Feature processing of content data

特征名称	特征样例	处理方式	结果样例
是否匿名	0、1	不处理	0、1
优质答案	0、1	不处理	0、1
是否推荐	0、1	不处理	0、1
创建时间	1540475871	截取前 4 位	1540
是否有图	0、1	不处理	0、1
感谢次数	63、655	等频划分	1、10
赞同次数	48、190	等频划分	1、4
搜藏次数	43、152	等频划分	1、4
反对次数	3、42	等频划分	1、5
举报次数	0、5	不处理	0、5

3.3　评价指标

在推荐过程中，若推荐数量过少，将会导致推荐结果的偶然性，使其不具有说服力；若推荐数量过多，由于每个用户浏览的时间不同，展示的项目数量不同，为浏览时间较短的用户进行推荐的项目无法全部展示，其结果将会产生误差。因此在评测中，根据用户平均浏览数量，选择 10 条项目作为推荐结果。

在线评测的评价标准计算公式如下：

$$\text{score}=\frac{N_{click}}{N_{online}} \tag{6}$$

其中：N_{click} 是用户点击量；N_{online} 是当天 10 000 名用户的在线数量。同时在 12 d 的推荐结果中选出最高的 8 d 得分，然后计算平均分。

3.4　实验结果与分析

1）XGBoost 参数调整

XGBoost 模型的构造十分简单，但若提高模型的效果，则需要进行参数的调整。XGBoost 模型为用户提供了多类参数，并且提供了便捷的 CV 函数供用户进行调参。在模型训练过程中，将用户的历史交互数据进行分割，80%作为训练集，20%作为测试集。然后使用 CV 函数得到最优参数。参

数如表 3 所示。

2）基于用户的协同过滤方法分析

在数据集中，用户浏览的每条内容都对应着多个话题类型，不同的浏览内容对应的话题可能相同，也可能不同。因此，在计算用户相似度时，既可以根据浏览的内容进行相似度的计算，又可以根据浏览内容所属的话题进行相似度计算。本实验分别使用基于内容的用户相似度和基于话题的用户相似度进行内容召回，然后使用相同的 XGBoost 模型进行计算，最后在知乎线上环境进行评测，其测试结果如表 4 所示。

表 3 XGBoost 模型参数

Table 3 Xgboost model parameters

参数	值
learning_rate	0.1
n_estimators	500
max_depth	11
min_child_weight	1
gamma	0
subsample	1
colsample_bytree	0.8
nthread	4
scale_pos_weight	1
seed	27
objective	binary:logistic
num_round	200

表 4 基于内容和基于话题的用户相似度对比

Table 4 Content-based and topic-based user similarity comparison

相似度	阈值	得分	平均分
基于话题的用户相似度	0.3	1.78 1.757	1.7685
基于内容的用户相似度	0.1	1.827 1.916	1.8715

在实验过程中，基于话题的用户相似度计算的召回率更高，因此设置阈值为 0.3，可以为大部分用户进行内容召回；基于内容的用户相似度计算的召回率较低，较小的阈值才可以为大部分用户进行内容的召回。使用同样的 XGBoost 模型进行预测，将预测结果推荐给用户。由结果显示，基于话题的用户相似度可以提高召回率，但同时会导致准确率降低。因此，基于内容的用户相似度在评测中得分更高，点击率也更高。

3）执行效率对比分析

在数据集中，若仅仅使用 XGBoost 算法，则需要在全部候选集中进行预测。当使用协同过滤进行粗粒度召回时，则可以减少模型的计算量。本实验通过使用 XGBoost 算法和融合协同过滤的 XGBoost 算法进行对比，当最大树深和迭代次数分别为 5、50 和 11、200 时，模型准确率最高。实验结果如图 2 所示。

由图 2 的评测结果显示，在全部候选集中进行预测，将会消耗大量的时间，融合协同过滤的 XGBoost 算法，由于协同过滤的召回，减少了模型的计算量，提高了模型的执行效率。

4） 模型效果对比分析

由式（2）可知，基于话题的用户相似度与基于内容的用户相似度相比，后者具有更好的推荐效果。使用式（1）（2）得到的结论，然后使用协同过滤、XGBoost 算法、协同过滤与 XGBoost 相融合的三种算法分别进行内容推荐，其在线上的评测结果如图 3 所示。

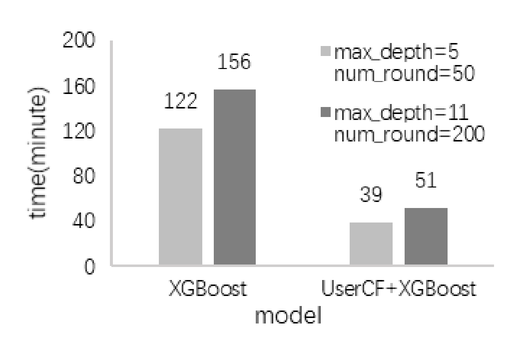


图 2 执行效率对比

Fig. 2 Comparison of execution efficiency

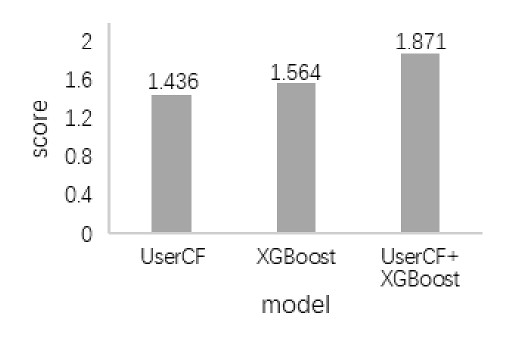


图 3 模型融合结果对比

Fig. 3 Comparison of model fusion results

模型融合结果对比如图 3 所示。由图 3 的评测结果显示，融合协同过滤的 XGBoost 推荐算法具有更高的点击率，能够提供更好的推荐效果。在实验过程中，由于协同过滤具有冷启动问题，部分用户的历史交互记录很少甚至没有，所以无法为这部分用户进行推荐。而对于 XGBoost 推荐算法，其需要对全部候选集进行计算，从而导致计算量的增加，耗时增多；同时候选集内容过多也会导致准确率的降低。融合协同过滤算法可以增加准确率，降低计算量，XGBoost 算法可以解决协同过滤的冷启动问题，同时可以增加准确率，因此模型融合的效果比单个模型的效果更好。

5） 评测结果对比分析

本次评测总共进行 12 次结果提交，选出 8 次评测最高分，计算出平均分作为最终结果。部分获奖选手的最终成绩如图 4 所示。

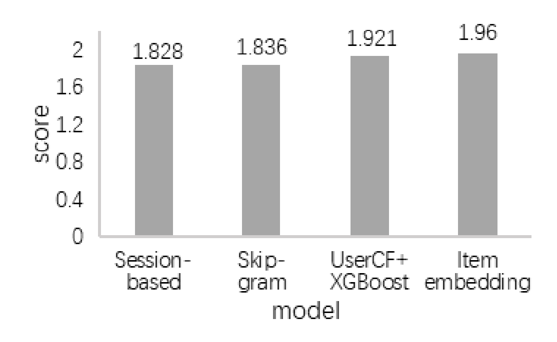


图 4 部分获奖用户评测得分

Fig. 4. Evaluation score of partial award-winning users

由图 4 的评测结果显示，与基于会话的推荐算法和基于 Skip-gram 推荐算法相比，融合协同过滤的 XGBoost 推荐算法具有较好的推荐效果；与基于项目嵌入的推荐算法相比，推荐效果略差一些。在线上评测环境中，不同选手分配的推荐用户群体不同，在评测得分上存在不确定因素，同时在 12 d 的评测中，模型或者参数的调整也会直接影响最终的评测

得分。但是真实的线上评测环境能够直接反映模型的推荐效果，与使用离线数据集相比，其结果更具有参考价值。

4　结束语

本文主要针对推荐系统中用户的冷启动问题进行改进，将协同过滤算法和 XGBoost 算法进行融合，首先使用协同过滤方法进行内容的粗粒度召回，然后使用 XGBoost 算法进行精确召回。本算法解决了用户冷启动问题，提高了模型的执行效率，提升了模型的预测能力，能够为用户准确推荐项目。同时，本文也存在不足之处，在模型的训练过程中未使用用户历史交互的时间序列特征。因此，下一步的工作将在模型中加入用户历史交互的时间序列特征，使模型达到更优的推荐效果。

参考文献：

[1] Marz N, Warren J. Big data: principles and best practices of scalable realtime data systems [M]//Greenwich: Manning. 2015.

[2] Yang Bo, Lei Yu, Liu Jiming, *et al.* Social collaborative filtering by trust [J]. IEEE Trans on Pattern Analysis and Machine Intelligence, 2017, 39 (8): 1633-1647.

[3] Al-Shamri M Y H. Power coefficient as a similarity measure for memory-based collaborative recommender systems [J]. Expert Systems with Applications, 2014, 41 (13): 5680-5688.

[4] 庞海龙，赵辉，李万龙，等. 融合协同过滤的线性回归推荐算法 [J/OL]. 计算机应用研究，2019 (5): 1-3. [2018-12-10]. http://www.arocmag. com/article/02-2019-05-004. html. (Pang Hailong, Zhao Hui, Li Wanlong, *et al.* Linear regression recommendation algorithm with collaborative filtering [J/OL]. Application Research of Computers, 2019 (5): 1-3. [2018-12-10]. http://www.arocmag.com/article/02-2019-05-004. html.)

[5] Wang Zhiqiang, Liang Jiye, Li Ru, *et al.* An approach to cold-start link prediction: establishing connections between non-topological and topological information [J]. IEEE Trans on Knowledge and Data Engineering, 2016, 28 (11): 2857-2870.

[6] 陈克寒，韩盼盼，吴健. 基于用户聚类的异构社交网络推荐算法 [J]. 计算机学报，2013, 36 (2): 349-359. (Chen Kehan, Han Panpan, Wu Jian. User clustering based social network recommendation [J]. Chinese Journal of Computers, 2013, 36 (2): 349-359.)

[7] 于洪，李俊华. 一种解决新项目冷启动问题的推荐算法 [J]. 软件学报，2015, 26 (6): 1395-1408. (Yu Hong, Li Junhua. Algorithm to solve the cold-start problem in new item recommendations [J]. Journal of Software, 2015, 26 (6): 1395-1408.)

[8] Pereira A L V, Hruschka E R. Simultaneous co-clustering and learning to address the cold start problem in recommender systems [J]. Knowledge-Based Systems, 2015, 82: 11-19.

[9] Shambour Q, Lu Jie. An effective recommender system by unifying user and item trust information for B2B applications [J]. Journal of Computer and System Sciences, 2015, 81 (7): 1110-1126.

[10] Kluver D, Ekstrand M D, Konstan J A. Rating-based collaborative filtering: algorithms and evaluation [M]// Social Information Access. Cham: Springer, 2018: 344-390.

[11] Park D H, Kim H K, Choi I Y, *et al.* A literature review and classification of recommender systems research [J]. Expert Systems with Applications, 2012, 39 (11): 10059-10072.

[12] Torre E, Quaglio P, Denker M, *et al.* Synchronous spike patterns in macaque motor cortex during an instructed-delay reach-to-grasp task [J]. Journal of Neuroscience, 2016, 36 (32): 8329-8340.

[13] Pirlo G, Impedovo D. Cosine similarity for analysis and verification of static signatures [J]. Iet Biometrics, 2013, 2 (4): 151-158.

[14] Chen Tianqi, Guestrin C. Xgboost: a scalable tree boosting system [C]// Proc of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York: ACM Press, 2016: 785-794.

[15] 李航. 统计学习方法 [M]. 北京：清华大学出版社，2012. (Li Hang. Statistical learning methods [M]. Beijing: Tsinghua University, 2012.)

[16] Kleinberg J, Lakkaraju H, Leskovec J, *et al.* Human decisions and machine predictions [J]. Nber Working Papers, 2018, 133 (1): 237-293.